



Bancos de Dados para Apps

SQLite

O SQLite pode ser definido como uma biblioteca de processo que implementa um banco de dados SQL autocontido, sem configuração, sem servidor e motor transacional. No caso do seu código-fonte está disponibilizado em domínio público e neste caso é gratuito para fins privados e comerciais. SQLite tem ligações a várias linguagens de programação como C, C++, BASIC, C#, Python, Java e Delphi. O COM (ActiveX) wrapper que torna o SQLite mais acessível principalmente a linguagens de script no Windows, também como VB Script e JavaScript, adicionando assim muitos mais recursos para aplicativos HTML. Também está disponível em sistemas operacionais incorporados, como iOS, Android, Symbian OS,

Maemo, Blackberry e WebOS devido ao seu pequeno tamanho e facilidade de uso.

SQLite é uma biblioteca de processo que implementa um mecanismo de banco de dados SQL. O código para SQLite é de domínio público e portanto, livre para uso para qualquer finalidade, comercial ou privada. SQLite é atualmente encontrado em mais aplicativos, incluindo vários projetos de alto nível.

SQLite é um mecanismo de banco de dados SQL incorporado e não possui um processo de servidor separado como a maioria dos outros bancos de dados SQL. Ele lê e grava diretamente em arquivos de disco comuns. O formato do arquivo de banco de dados é

multiplataforma. Assim, esses recursos irão tornar o SQLite uma escolha popular e com formato de arquivo de aplicativo. SQLite é uma biblioteca compacta, o tamanho da biblioteca pode ser menor que 500 KiB, dependendo da plataforma de destino e das configurações de otimização do compilador. Se os recursos opcionais forem omitidos, o tamanho da biblioteca SQLite pode ser reduzida abaixo de 300 KiB. O SQLite também pode ser executado em espaço mínimo de pilha (4KiB).

A maior parte do código-fonte SQLite é dedicada exclusivamente a testes e verificação. O SQLite responde normalmente a falhas de alocação de memória e erros de E/S de disco. É claro que, mesmo com todos esses testes, ainda há problemas. Mas o SQLite é aberto e honesto sobre todos os bugs e fornece listas de bugs incluindo listas de bugs críticos e cronologias minuto a minuto de relatórios de bugs e mudanças de código.

Segue um modelo da arquitetura do SQLite abaixo:

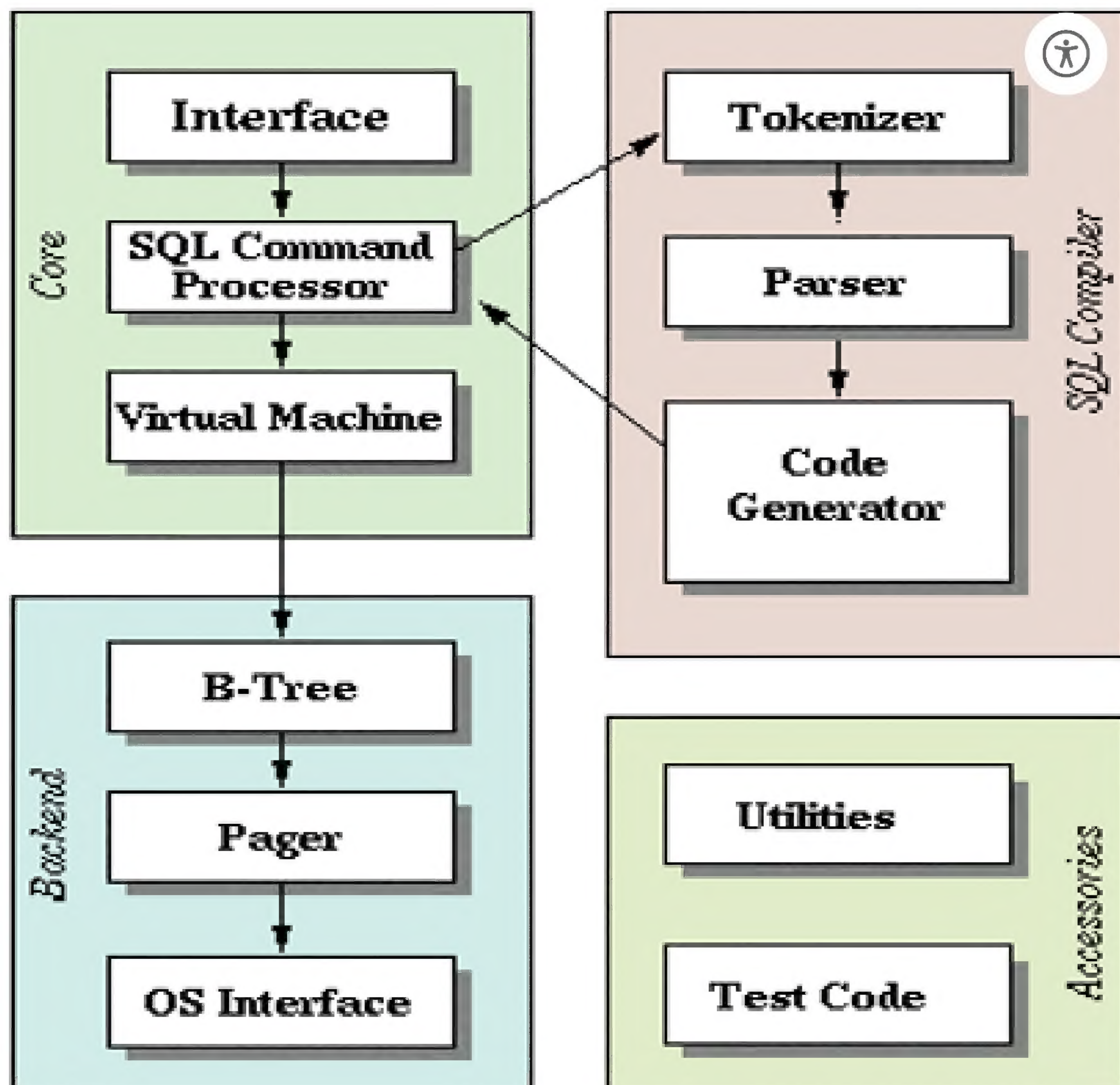


Figura 1. Arquitetura SQLite

A arquitetura do SQLite é relativamente simples, ela é composta por quatro partes, 1- Core, a parte principal contém a interface do usuário, o processador de comandos SQL e a máquina virtual. 2- SQL compiler, contém um tokenizer, um analisador e um gerador de código. 3- Backend, contém B-Tree, Page Cache, OS Interface: Um banco de dados SQLite é mantido em disco usando uma B-tree implementação encontrada no arquivo fonte btree.c. 4- Acessórios contém Utilitários e código de teste: SQLite fornece algumas

funcionalidades relacionadas à utilitários, como memória alocação e rotinas de comparação de strings sem maiúsculas e minúsculas estão localizadas em util.c mais da metade da base de código total do SQLite é dedicado a testes.

Podemos demonstrar, com um pequeno exemplo a criação de uma agenda telefônica utilizando o SQLite:

Criando a tabela agenda:

```
CREATE TABLE agenda(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(30),
  tel VARCHAR(20)
);
```

Uma forma interessante de guardar todos os comandos digitados é criar um arquivo com extensão .sql e depois é só entrar no ambiente e digitar o comando .read arquivo.sql, que ele irá abrir todo o conteúdo do arquivo. Vamos mostrar como podemos inserir alguns dados na minha tabela agenda criada:

```
INSERT INTO agenda(nome,tel) VALUES ('Central de Transplantes', '0800-8832323');
INSERT INTO agenda(nome,tel) VALUES ('Disque-Silencio', '3452-6927');
INSERT INTO agenda(nome,tel) VALUES ('Bombeiros', '193');
INSERT INTO agenda(nome,tel) VALUES ('Ambulância', '192');
INSERT INTO agenda(nome,tel) VALUES ('Policia Militar', '190');
INSERT INTO agenda(nome,tel) VALUES ('Defesa Civil', '199');
INSERT INTO agenda(nome,tel) VALUES ('Disque Cidade Limpa','0800-851531');
```

Já com algumas linhas inseridas, agora podemos fazer algumas consultas na nossa tabela:



```
$ sqlite telefones.db
SQLite version 2.8.17
Enter ".help" for instructions
sqlite> .read telefones.sql
sqlite> SELECT * FROM agenda;
1|Central de Transplantes|0800-8832323
2|Disque-SilÃªncio|3452-6927
3|Bombeiros|193
4|Ambulancia|192
5|Policia Militar|190
6|Defesa Civil|199
7|Disque-Cidade Limpa|0800-851531
```

Agora vamos criar nosso modelo de banco de dados para cadastro dos postos de gasolina:



```
CREATE TABLE posto(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(30),
  tel VARCHAR(20),
  latitude VARCHAR(40),
  longitude VARCHAR(40),
);
```

```
CREATE TABLE Combustiveis(
 id INTEGER PRIMARY KEY AUTOINCREMENT,
 posto_id INTEGER,
 Name TEXT NOT NULL,
 Price NUMERIC,
 FOREIGN KEY(posto_id) REFERENCES posto(id)
);
```

```
CREATE TABLE usuario(
  id INTEGER PRIMARY KEY,
  nome TEXT NOT NULL,
  telefone TEXT NOT NULL,
  password TEXT NOT NULL,
  email TEXT NOT NULL,
);
```

Firebase

O Firebase é considerado uma plataforma de aplicação w sso ajuda os desenvolvedores a criarem aplicativos de alta qualidade. Ele armazena os dados em Formato JavaScript Object Notation (JSON) que não usa consulta para inserir, atualizar, excluir ou adicionar dados a ele. Isso é o backend de um sistema que é usado como banco de dados para armazenar dados.

Os serviços que são disponibilizados pelo firebase são:

Firebase Analytics

Ele fornece informações sobre o uso do aplicativo. É um aplicativo pago de solução de medição que também proporciona engajamento do usuário.

Firebase Cloud Messaging (FCM)

Anteriormente conhecido como Google Clouds Messaging (GCM), o FCM é um serviço pago que é uma solução multiplataforma para mensagens e notificações para Android, aplicativos da Web e IOS.

Firebase Auth

O Firebase Auth oferece suporte a provedores de login social como o Facebook, Google GitHub e Twitter. É um serviço que pode autenticar usuários usando apenas código do lado do cliente e é pago pelo serviço.

Real-time Database

O Firebase fornece serviços como banco de dados em tempo real e processo interno. Uma API é fornecida ao desenvolvedor do

aplicativo que permite que os dados do aplicativo sejam sincronizados entre clientes e armazenados na Firebase Clouc

Firebase Storage

Facilita a transferência de arquivos fácil e segura, independentemente da rede, da qualidade para os aplicativos do Firebase.

Firebase Test Lab for Android

Ele fornece infraestrutura baseada em nuvem para testar o Android apps. Com uma operação, os desenvolvedores podem iniciar o teste de seus aplicativos em uma ampla variedade de dispositivos e configurações.

Firebase Crash Reporting

Os relatórios detalhados dos erros são criados no aplicativo. Os erros são agrupados em clusters de rastreamentos de pilha semelhantes e triados pela gravidade. Os outros recursos são: o desenvolvedor pode registrar eventos personalizados para ajudar a capturar as etapas que levam a um colidir.

Firebase Notifications

Ele permite notificações de usuários direcionadas para aplicativos móveis

desenvolvedores e os serviços estão disponíveis gratuitamente.

Construindo um Banco de Dados SQLite IONIC

Agora vamos criar um projeto IONIC e adicionar a biblioteca SQLite com o seguinte comando:

ionic start pratica-integradora tabs --type=angular –capacitor

Adicionar a biblioteca de SQLite:

npm install cordova-sqlite-storage

npm install @awesome-cordova-plugins/sqlite

ionic cap sync

Vamos alterar o arquivo app.module.ts

```
import { SQLite } from '@awesome-cordova-plugins/sqlite/ngx';

@NgModule({

  declarations: [AppComponent],

  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],

  providers: [SQLite,

    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],

  bootstrap: [AppComponent],

})

export class AppModule {}
```

Agora vamos alterar o componente principal e o arquivo app.component.ts



```
import { Component } from '@angular/core';

import { SQLiteObject } from '@awesome-cordova-plugins/sqlite'

import { SQLite } from '@awesome-cordova-plugins/sqlite/ngx'

import { Platform } from '@ionic/angular';

const SQL    'CREATE TABLE IF NOT EXISTS user(id INTEGER PRIMARY KEY, name TEXT NOT NULL,email TEXT NOT NULL,'

 'phone TEXT NOT NULL,password TEXT NOT NULL);'

 'CREATE TABLE IF NOT EXISTS gas_station ( id INTEGER PRIMARY KEY, title TEXT NOT NULL,'+

 'address TEXT NOT NULL, address TEXT NOT NULL, number TEXT NOT NULL, city TEXT NOT NULL,lat TEXT NOT NULL, lng TEXT NOT NULL);'+

 'CREATE TABLE IF NOT EXISTS fuel (id INTEGER PRIMARY KEY AUTOINCREMENT, posto_id INTEGER,'+

     'Name TEXT NOT NULL,Price NUMERIC,FOREIGN KEY(gas_station_id) REFERENCES gas_station(id));'

@Component({

 selector: 'app-root',

 templateUrl: 'app.component.html',

 styleUrls: ['app.component.scss'],
```



```
});

export class AppComponent {

 constructor(private SQLite public Platform

  if(this.platform.is('android')){ this

 }

  this.sqlite.create({ name: 'data.db', location: 'default' })

   ((db: SQLiteObject) => { db.executeSql(SQL

   (() => console.log('Executed SQL'))

   (e => console.log(e));

  })(e => console.log(e));

 }

}
```

Posteriormente vamos aprofundar mais sobre o SQLite a gravar informações no banco de dados diretamente no Android.

Ferramentas para Administrar Banco de Dados

Quando falamos de ferramentas de banco de dados temos que saber que essas ferramentas precisam além de vencer as barreiras das redes internas ou mesmo os acessos ao banco de dados de

fornecedores diversos, também terão que auxiliar os administradores de maneiras específicas desempenhar as tarefas como gerenciamento do espaço utilizado pelo banco de dados, fazer uma análise do desempenho do mesmo e também permitir o fácil e rápido acesso aos dados dos principais tipos de objeto que ele possui.

Mas sabemos que a tarefa de administrar um banco de dados vai muito além de tudo isso, que irão exigir uma capacidade grande de análise e envolvimento do administrador do banco de dados. Exatamente por esse motivo uma boa ferramenta de banco de dados precisa ter a capacidade de fornecer para esse administrador uma área onde o mesmo poderá desenvolver seus conhecimentos através de comandos livres que irão fazer a extração das informações que servirão para a sua avaliação e para a solução dos problemas por parte do banco de dados.

Agora vamos mostrar algumas ferramentas que irão administrar o banco de dados:

MySql Workbench

O MySql workbench é uma ferramenta visual de design de banco de dados, onde o grande objetivo é fazer uma integração do desenvolvimento de banco de dados com o design, a criação e a manutenção SQL tudo isso dentro de um único ambiente de desenvolvimento. É um ambiente de multiplataforma e desenvolvido pelo Oracle, ele possui a licença GNU (General Public License) e se coloca como uma excelente oportunidade de substituir o ambiente de terminal.

Podemos destacar as seguintes vantagens:

- É um ambiente multiplataforma;

- Possui uma excelente documentação;



- Faz uma conexão direta com o banco de dados;
- Sua versão é distribuída de forma gratuita;
- Além de possuir uma curva elevada de aprendizado e com certeza outras diversas vantagens.

PhpMyAdmin

O phpMyAdmin é uma ferramenta de software livre escrita em PHP, destinada a lidar com a administração do MySQL pela Web. O phpMyAdmin suporta uma ampla variedade de operações no MySQL e no MariaDB. As operações usadas com frequência (gerenciamento de bancos de dados, tabelas, colunas, relações, índices, usuários, permissões, etc.) podem ser realizadas através da interface do usuário, enquanto você ainda tem a capacidade de executar diretamente qualquer instrução SQL.

O phpMyAdmin vem com uma ampla variedade de documentação e os usuários são bem-vindos para atualizar nossas páginas wiki para compartilhar ideias e instruções para várias operações.

Para facilitar o uso para uma ampla gama de pessoas, o phpMyAdmin está sendo traduzido para 72 idiomas e suporta idiomas LTR e RTL.

O phpMyAdmin é um projeto maduro com uma base de código estável e flexível.

O projeto phpMyAdmin é membro da Software Freedom Conservancy. A SFC é uma organização sem fins lucrativos que ajuda a promover, melhorar, desenvolver e defender projetos de Software Livre, Livre e de Código Aberto (FLOSS).

Seus recursos são:

- Interface web intuitiva;
- Suporte para a maioria dos recursos do MySQL: navegar e soltar bancos de dados, tabelas, visualizações, campos e índices, criar, copiar, descartar, renomear e alterar bancos de dados, tabelas, campos e índices servidor de manutenção, bancos de dados e tabelas, com propostas de configuração do servidor;
- Execute, edite e marque qualquer instrução SQL, mesmo consultas em lote;
- Gerenciar contas e privilégios de usuário MySQL;
- Gerenciar procedimentos armazenados e gatilhos;
- Importar dados de CSV e SQL;
- Exporte dados para vários formatos: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX e outros;
- Administrando vários servidores;
- Criando gráficos do layout do seu banco de dados em vários formatos;

- Criando consultas complexas usando Query-by-example (QBE);
- Pesquisando globalmente em um banco de dados ou em um subconjunto dele;
- Transformando dados armazenados em qualquer formato usando um conjunto de funções predefinidas, como exibir dados BLOB como imagem ou link de download.

TablePlus

O TablePlus é uma ferramenta nativa moderna com interface de usuário elegante que permite gerenciar simultaneamente vários bancos de dados, como MySQL, PostgreSQL, SQLite, Microsoft SQL Server e muito mais.

Atualmente você pode usar o TablePlus para trabalhar com:

PostgreSQL, MySQL, MariaDB, SQLite, Microsoft SQL Server, Amazon Redshift, Oracle (somente macOS), CockroachDB, Floco de neve (macOS e Windows), Cassandra, Redis, Vertica e MongoDB (Beta). E qualquer banco de dados que use o mesmo protocolo com os bancos de dados listados acima, como Percona, Yugabyte DB, etc. O TablePlus nunca envia ou armazena nenhum de seus dados no servidor. Todas as credenciais do banco de dados são mantidas com segurança no armazenamento rígido do dispositivo do usuário.

O TablePlus armazena o endereço de e-mail do cliente no servidor nos EUA apenas para o gerenciador de licenças. A identificação do dispositivo é um hash unidirecional e não contém nenhuma informação privada sobre o dispositivo do usuário, apenas um hash.

O TablePlus usa o AppCenter, um produto da Microsoft para o relatório de falhas. O período de retenção de dados atual é de 90 dias. No entanto, você pode desativá-lo na preferência do aplicativo.

Atividade Extra

Para se aprofundar no assunto desta aula, assista o vídeo de Modelagem de dados com Mysql Workbench, de Adriano Souza.

Canal: AdrianoSouza

Título a ser buscado no youtube: **Modelagem de dados com Mysql Workbench 1 baixando**

Referência Bibliográfica

STONEHEM, Bill. Google Android Firebase: **Learning the Basics Paperback**, 2016 datado em 18/3/17.

KALSOV, 2012. **Developer Meet Firebase** data 10/05/2022.

IONIC Framework. **IONIC**. Disponível em:

<https://ionicframework.com/docs/native/sqlite>. Acesso em: 30 nov. 2022.

Cordova SQLite. **SQLite Storage**. Disponível em:

<https://github.com/storesafe/cordova-sqlite-storage>. Acesso em: 30 nov. 2022.

Atividade Prática 1 - Banco de Dados

Título da Prática: Bancos de Dados para Apps

Objetivos: Utilizar o banco de dados SQLite para fazer um projeto de banco de dados.

Materiais, Métodos e Ferramentas: Iremos fazer uma pesquisa na internet e utilizar o SGBD SQLite para fazer as construções.

Atividade Prática



O SQLite pode ser definido como uma biblioteca de processo que implementa um banco de dados SQL autocontido, sem configuração, sem servidor e motor transacional. No caso do seu código-fonte está disponibilizado em domínio público e neste caso é gratuito para fins privados e comerciais. SQLite tem ligações a várias linguagens de programação como C, C++, BASIC, C#, Python, Java e Delphi. O

COM (ActiveX) wrapper que torna o SQLite mais acessível principalmente a linguagens de script no Windows, também como VB Script e JavaScript, adicionando assim muitos mais recursos para aplicativos HTML. Também está disponível em sistemas operacionais incorporados, como iOS, Android, Symbian OS, Maemo, Blackberry e WebOS devido ao seu pequeno tamanho e facilidade de uso.

SQLite é uma biblioteca de processo que implementa um mecanismo de banco de dados SQL. O código para SQLite é de domínio público e portanto, livre para uso para qualquer finalidade, comercial ou privada. SQLite é atualmente encontrado em mais aplicativos, incluindo vários projetos de alto nível.

SQLite é um mecanismo de banco de dados SQL incorporado e não possui um processo de servidor separado como a maioria dos outros banco de dados SQL. Ele lê e grava diretamente em arquivos de disco comuns. O formato do arquivo de banco de dados é multiplataforma. Assim, esses recursos irão tornar o SQLite uma escolha popular e com formato de arquivo de aplicativo. SQLite é uma biblioteca compacta, o tamanho da biblioteca pode ser menor que 500KiB, dependendo da plataforma de destino e das configurações de otimização do compilador. Se os recursos opcionais forem omitidos, o tamanho da biblioteca SQLite pode ser reduzida abaixo de 300KiB. O SQLite também pode ser executado em espaço mínimo de pilha (4KiB).

A maior parte do código-fonte SQLite é dedicada exclusivamente a testes e verificação. O SQLite responde normalmente a falhas de alocação de memória e erros de E/S de disco. É claro que, mesmo com todos esses testes, ainda há problemas. Mas o SQLite é aberto e honesto sobre todos os bugs e fornece listas de bugs incluindo

listas de bugs críticos e cronologias minuto a minuto de relatórios de bugs e mudanças de código.

A proposta dessa atividade prática é fazer a criação de um modelo de banco de dados baseado no SQLite, para isso vamos colocar abaixo um exemplo de criação de uma tabela para ter um modelo:

Criando a tabela agenda:

```
CREATE TABLE agenda(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(30),
  tel VARCHAR(20)
);
```

Vamos a proposta de criação do modelo de banco de dados: "Um berçário deseja informatizar suas operações. Quando um bebê nasce, algumas informações são armazenadas sobre ele, tais como: nome, data do nascimento, peso do nascimento, altura, a mãe deste bebê e o médico que fez seu parto. Para as mães, o berçário também deseja manter um controle, guardando informações como: nome, endereço, telefone e data de nascimento. Para os médicos, é importante saber: CRM, nome, telefone celular e especialidade".

A partir desse cenário, crie as tabelas, seus atributos, suas relações, faça as inserções devidas e crie pelo menos 5 consultas nas tabelas.

Gabarito Atividade Prática



Para responder esta questão é necessário primeiramente verificar os modos de criação do banco de dados usando o SQLite, como criação de tabelas, inserção de dados, criação de relacionamentos entre duas ou mais tabelas usando o join e desenvolver algumas consultas utilizando o select. Ao final colocar os códigos para a verificação.

// nome, data do nascimento, peso do nascimento, altura, a mãe deste bebê e o médico

```
CREATE TABLE bebes(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(150),
  data_nascimento DATE,
  peso NUMBER,
              altura NUMBER,
 mae_id INTEGER,
 medico_id INTEGER
);
// Para as mães, o berçário também deseja manter um controle, guardando informações como: nome, endereço, telefone e data de nascimento.

CREATE TABLE maes(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(150),
  data_nascimento DATE,
  endereco VARCHAR(255),
  telefone VARCHAR(20)
);
```

// Para os médicos, é importante saber: CRM, nome, telefone cel... e especialidade

```
CREATE TABLE medicos(
  id INTEGER PRIMARY KEY,
  nome VARCHAR(30),
  CRM  VARCHAR(30),
  endereco VARCHAR(255),
  telefone VARCHAR(20),
            especialidade VARCHAR(200)

);

SELECT * FROM BEBES WHERE ID = 1 INNER JOIN MEDICOS id = medico_id;
```

Ir para exercício